

# UNIVERSITÀ DELLA CALABRIA



---

DIPARTIMENTO DI MATEMATICA E INFORMATICA

Corso di Laurea Triennale in Informatica

Tesi di Laurea Triennale

## Ontology Mapping con ASP

**RELATORE**

*Prof. Francesco Ricca*

**CANDIDATO**

*Amato Davide*

Matricola 163876

**CORRELATORE**

*Dott. Bernando Cuteri*

---

ANNO ACCADEMICO 2016/2017

Alla mia famiglia...

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Preliminari</b>	<b>6</b>
2.1	RDF . . . . .	6
2.2	OWL . . . . .	9
2.3	Ontologie . . . . .	9
2.4	CIDOC-CRM . . . . .	12
2.5	Ontology Mapping . . . . .	12
2.6	Answer Set Programming (ASP) . . . . .	13
<b>3</b>	<b>Ontology Mapping con ASP</b>	<b>15</b>
3.1	Procedimento . . . . .	15
<b>4</b>	<b>Validazione</b>	<b>21</b>
4.1	Applicazione CHQA . . . . .	21
4.2	Validazione tramite CHQA . . . . .	22
<b>5</b>	<b>Conclusioni</b>	<b>24</b>

# Capitolo 1

## Introduzione

**Contesto** Dalla sua nascita, il Web si sta evolvendo costantemente. Quando Tim Berners-Lee lo creò (1990), insieme a Robert Cailliau, lo fece con un obiettivo:

*Il Web è più un'innovazione sociale che un'innovazione tecnica. L'ho progettato perché avesse una ricaduta sociale, perché aiutasse le persone a collaborare, e non come un giocattolo tecnologico.*[2, p. 113].

Inoltre auspicava un'evoluzione del Web dove quest'ultimo sarebbe diventato uno strumento che avrebbe migliorato enormemente la connessione tra i popoli e reso estremamente intuitiva e immediata la reperibilità, fruibilità e condivisione del sapere.

Al momento il Web potrebbe essere considerato come un semplice database, pieno di contenuti di qualsiasi tipo, accessibili esclusivamente tramite collegamenti diretti (i *link*). Questa visione di Web, però, è destinata a cambiare in modo radicale, tutto grazie al “*Semantic Web*”.

Immaginate un futuro dove le tipiche ricerche sul Web per parole chiave sono solo un brutto ricordo. Un futuro dove i computer non solo hanno libero accesso a tutti i dati presenti sul Web, ma soprattutto riescono a comprenderne il significato e le relazioni che li collegano.

Quando si parla di *Semantic Web* si intende proprio quest'evoluzione del Web in un ambiente dove i documenti pubblicati (pagine HTML, file, immagini, e così via) sono interconnessi tramite relazioni dettate dalla semantica degli

stessi.

In quest'ambito, hanno grande rilevanza le “*ontologie*”. Un'ontologia può essere vista, in modo molto semplicistico, come un dizionario dove sono descritti i concetti e le relazioni esistenti all'interno di un determinato dominio. Queste vengono usate da applicazioni di semantic web per svariati scopi. A volte, però, queste applicazioni hanno bisogno di “far comunicare” due o più ontologie tra di loro. Per fronteggiare questo problema si può ricorrere all'*Ontology Mediation* [1].

L'*Ontology Mediation* è quell'insieme di processi atti a permettere il riuso dei dati fra le diverse applicazioni sul Semantic Web e la condivisione degli stessi tra basi di conoscenza eterogenee.

Tra questi processi, sono degni di menzione:

- Ontology Mapping
- Ontology Alignment
- Ontology Merging

**Ontology Mapping** Consiste nel trovare le corrispondenze tra le ontologie. Queste corrispondenze vengono poi utilizzate per “tradurre” i dati tra le diverse rappresentazioni e sono espresse mediante assiomi formulati in uno specifico linguaggio di mapping.

**Ontology Alignment** Quando il processo di scoperta delle corrispondenze è (semi-)automatizzato mediante l'uso di algoritmi prende il nome di Ontology Alignment.

**Ontology Merging** Attività che ha lo scopo di creare una nuova ontologia dall'unione delle ontologie prese in input, così da unificarne la conoscenza. Ha l'obiettivo, quindi, di riflettere tutte le corrispondenze, ma anche le differenze, nell'ontologia unificata.

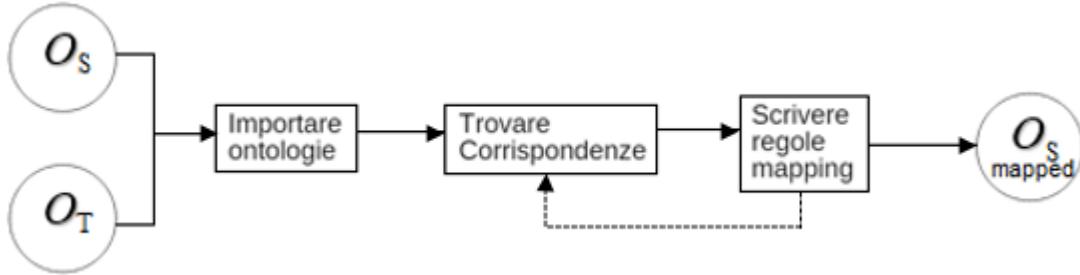


Figura 1.1: Processo di Ontology Mapping

La problematica principale da affrontare nell’Ontology Mediation sono gli *ontology mismatches* (disallineamenti ontologici), che si suddividono a loro volta in:

- disallineamenti concettuali (*conceptualization mismatches*).
- disallineamenti di rappresentazione (*explication mismatches*).

**Disallineamenti concettuali** Si parla di disallineamenti concettuali quando, ad esempio, ci sono delle differenze nella parte di dominio coperta da entrambe le ontologie o, ancora, quando ci sono differenze nel livello di dettaglio.

**Disallineamenti di rappresentazione** Si parla, invece, di disallineamenti di rappresentazione quando il modo in cui i concetti sono descritti differisce (“sottoclassi” contro “attributi”), quando ci sono concetti equivalenti ma rappresentati con nomi diversi o quando le ontologie sono codificate in maniera differente (“km” contro “miglia”).

In questa tesi, però, ci focalizzeremo sull’Ontology Mapping [Fig. 1.1].

**Scopo** Lo scopo di questo lavoro è dimostrare come le tecniche di “*Answer Set Programming*” (ASP) possano essere utilizzate per approcciare, in modo piuttosto semplice ed intuitivo, le problematiche relative all’*Ontology Mapping* nel contesto del *Semantic Web*.

**Struttura elaborato** L’elaborato si suddivide in tre parti:

- Nella prima parte (cap.2) verranno introdotti in modo sufficientemente esaustivo alcuni concetti che serviranno per la comprensione del lavoro svolto.
- Nella seconda (cap.3) verrà illustrato nel dettaglio il procedimento atto ad eseguire il “mapping” tra due ontologie. Quella di partenza è “DBpedia Italiana” (un’ontologia che contiene le informazioni estratte da Wikipedia in Italiano), mentre l’ontologia target è “CIDOC-CRM”, divenuta ormai uno standard riconosciuto nella rappresentazione dell’informazione riguardante i beni culturali.
- Nella terza parte (cap.4) verranno mostrati i risultati ottenuti tramite l’utilizzo del sistema di question answering, relativo al patrimonio culturale, “CHQA” sviluppato all’interno del progetto europeo “PiùCultura”, come strumento di validazione del lavoro svolto.

# Capitolo 2

## Preliminari

Di seguito verranno introdotti gli elementi principali che costituiscono le fondamenta del *Semantic Web* [Fig. 2.1], ovvero RDF, OWL, ontologie, e altre nozioni fondamentali per la comprensione dell'intero lavoro.

### 2.1 RDF

Quando si parla di Semantic Web, non si può fare a meno di menzionare il *Resource Description Framework* (RDF). L'RDF è l'elemento base che consente l'interoperabilità semantica tra applicazioni che condividono le informazioni sul Web.

Il modello RDF è formato da risorse, proprietà e valori. Per “risorsa” si intende qualsiasi cosa che si possa identificare con un *URI* (Uniform Resource Identifier), le “proprietà” sono delle relazioni che legano le risorse con i valori, mentre i “valori” sono dati primitivi (stringhe, numeri, ecc...). Da notare che un valore può essere anche una stringa contenente l'URI di una risorsa.

L'unità base per rappresentare un'informazione in RDF è lo *statement* [Fig. 2.2]. Uno *statement* è una tripla composta da *Soggetto – Predicato – Oggetto*, dove il soggetto è una risorsa, il predicato una proprietà e l'oggetto è un valore. Questa struttura non è altro che un grafo orientato dove i soggetti e gli oggetti sono nodi, mentre gli archi che li collegano sono i prediciati [Fig. 2.3].

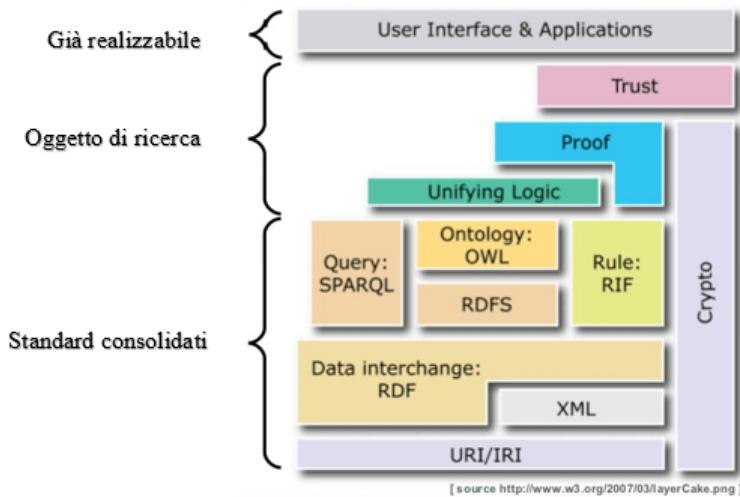


Figura 2.1: Rappresentazione grafica a strati del Semantic Web.

```

@prefix espers: <http://www.esempio.it/persona/> .
@prefix esi: <http://www.esempio.it/istituzione/> .
@prefix esl: <http://www.esempio.it/luogo/> .
@prefix esp: <http://www.esempio.it/predicato/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

espers:163876 esp:ha_nome "Amato Davide"^^xsd:string .
espers:163876 esp:ha_immagine espers:163876/img.jpg .
espers:163876 esp:studia_presso esi:unical .
esi:unical esp:ha_nome "Università della Calabria"^^xsd:string .
esi:unical esp:ha_logo esi:unical/logo.png .
esi:unical esp:è_situata_in esl:Rende_(CS) .
esl:Rende_(CS) esp:ha_nome "Rende"^^xsd:string .
esl:Rende_(CS) esp:ha_numero_abitanti "35586"^^xsd:integer .

```

Figura 2.2: Esempio di alcuni *statement* RDF (in formato Turtle con prefissi).

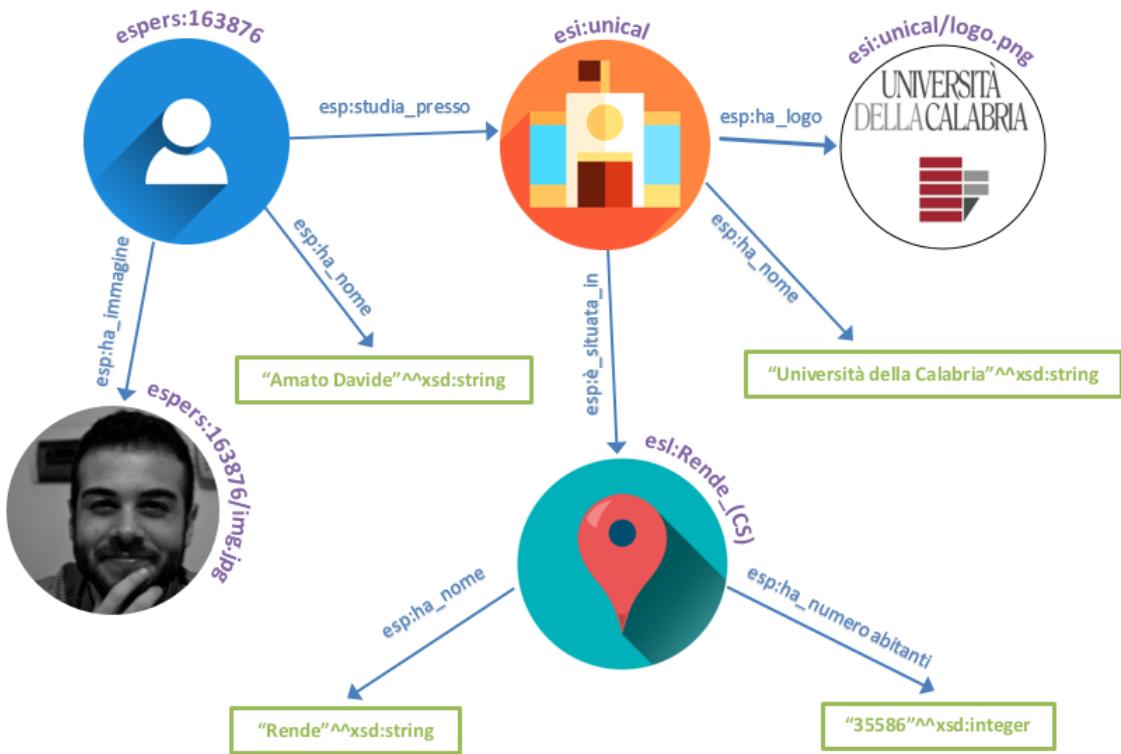


Figura 2.3: Rappresentazione delle informazioni contenute in Fig. 2.2 sotto forma di *grafo RDF*

## 2.2 OWL

Il *Web Ontology Language* (OWL) è un linguaggio di *markup* per rappresentare esplicitamente significato e semantica di termini con vocabolari e relazioni tra gli stessi.

OWL è un'estensione del RDF ed il suo scopo è descrivere delle basi di conoscenze, effettuare delle deduzioni su di esse e integrarle con i contenuti delle pagine web.

OWL intende rendere possibile, ad esempio:

- ricerche nel web che superino i problemi di omonimia e ambiguità presenti nelle normali ricerche testuali
- applicazioni che effettuino delle deduzioni sui dati

Questo linguaggio viene utilizzato per la creazione delle *ontologie*.

## 2.3 Ontologie

L'ontologia, in *lato sensu*, è quel ramo della filosofia che studia l'essere in quanto tale, nonché le sue categorie fondamentali.

In informatica, invece, col termine *ontologia* si intende la rappresentazione in modo formale ed esplicito dei concetti e delle relazioni che definiscono un particolare dominio. Il punto di forza delle ontologie sta proprio nella loro chiarezza e completezza. Sono applicate comunemente nel campo dell'intelligenza artificiale (ragionamento deduttivo, classificazione, tecniche di problem solving) e nella rappresentazione e condivisione della conoscenza, facilitando la comunicazione e lo scambio di informazioni fra diversi sistemi.

Le ontologie possono essere classificate sommariamente in due tipologie:

- *ontologie costitutive* (o superiori)
- *ontologie specializzate*

**ontologie costitutive** Non descrivono un particolare dominio, ma entità più generali ed universali. Offrono quindi dei concetti fondanti e delle assiomatizzazioni che possono poi essere utilizzate per creare le ontologie specializzate.

**ontologie specializzate** Sono ontologie che definiscono tutti i concetti e le relazioni utili a rappresentare le informazioni di un dominio specifico.

Inoltre può essere utile introdurre la nozione di *ontologia popolata*, ovvero un'ontologia che non contiene esclusivamente definizioni di concetti e relazioni di dominio, ma anche istanze di questi. Ne sono un chiaro esempio DBpedia e BabelNet.

Nel 2007 il W3C (*World Wide Web Consortium*), organizzazione fondata dallo stesso Berners-Lee con lo scopo di “*Guidare il Web fino al massimo del suo potenziale...*” (motto del W3C), fece partire un progetto chiamato *Linking Open Data Project*. L’obiettivo principale di questo progetto è quello di convertire in formato RDF i *data sets* presenti sul web, così da ampliare sempre di più questa rete di connessioni tra dati condivisi [Fig. 2.4]. Il numero di ontologie condivise sul web sta aumentando costantemente, così come il bisogno di applicazioni che utilizzino quest’ultime.

A volte può capitare, però, che un’applicazione utilizzi dapprima un’ontologia e, successivamente, necessiti di un’estensione tramite l’integrazione di un’altra ontologia.

Facciamo un esempio. Supponiamo di voler gestire un archivio multimediale. A supporto di quest’applicazione potremmo voler sviluppare un’ontologia che chiameremo “*MOnt*”. Questa dovrà definire tutti quei concetti (“Autore”, “Protagonista”, “Opera multimediale”, “Descrizione”, “Titolo”, “Durata”, “Genere”, ecc...) e relazioni (“è intitolata”, “è di tipo”, “è stata creata da”, ecc...) che definiscono il dominio delle produzioni multimediali. Successivamente vorremmo integrare nella nostra applicazione informazioni contenute in un data set (in formato RDF) relative ad opere cinematografiche descritte in un’ontologia dal nome fittizio “*CinemOnt*”. Nonostante il dominio sia affine a quello di MOnt, CinemOnt ha concetti e relazioni differenti (ad esempio potrebbe avere concetti del tipo “Comparsa”, “Regista”, “Trama”, e relazioni come “è stato girato da”, “ha trama”, “ha fatto un cameo in”).

Il problema da porci adesso è:

“*Come faremo ad integrare due ontologie?*”

Ed è qui che entra in gioco il processo denominato *Ontology Mapping*, processo che vedremo in dettaglio nel paragrafo 2.5.

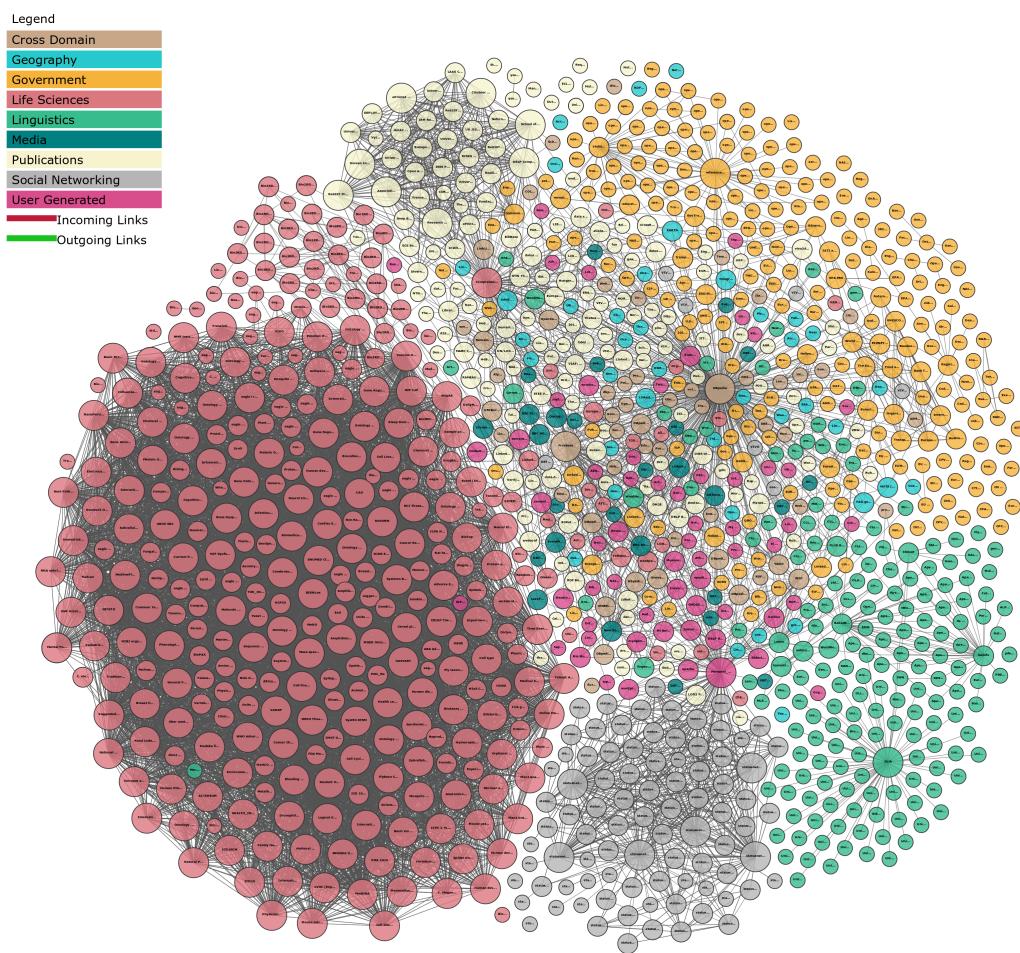


Figura 2.4: Linked Open Data Cloud - 2017 (Fonte: <http://lod-cloud.net>).

## 2.4 CIDOC-CRM

Il *CIDOC-CRM* (Conceptual Reference Model) è un modello concettuale orientato agli oggetti creato da *CIDOC*, che nel 2006 è diventato di fatto uno standard di riferimento nella rappresentazione e nello scambio di informazioni relative al patrimonio culturale (ISO 21127:2006, adesso ISO 21127:2014). Il CIDOC-CRM è un'*ontologia specializzata* (par. 2.3) che fornisce una serie di definizioni, concetti (impliciti ed esplicativi) e relazioni nella quale tutti i documenti relativi al patrimonio culturale possono essere mappati.

## 2.5 Ontology Mapping

L'*Ontology Mapping* è un processo attraverso il quale i concetti e le relazioni di un'ontologia vengono convertite, attraverso alcune “regole di mapping”, in concetti e relazioni di un'altra ontologia. Per una migliore comprensione, partiamo dalla definizione algebrica di ontologia [5]. Un'ontologia può essere considerata come una coppia  $O = (G, A)$  dove  $G$  è la *firma ontologica (ontological signature)* - ovvero il grafo  $G = (C, P)$  rappresentante l'insieme di concetti ( $C$ ) e predicati ( $P$ ) di cui è composta l'ontologia – ed  $A$  è l'insieme di assiomi contenuti in essa ( $A$  potrebbe essere vuoto).

Un esempio potrebbe essere [4]:

$$O = (G, A) \text{ con } G = (C, P)$$

dove:

- $C = \{ \text{Entità, Oggetto, Persona, Meccanico, Automobile, Motore} \}$  ;
- $P = \{ \text{è-un, ha-un, ripara} \}$  :
  - $\text{è-un} = \{ (\text{Oggetto}, \text{Entità}), (\text{Persona}, \text{Entità}), (\text{Meccanico}, \text{Persona}), (\text{Automobile}, \text{Oggetto}), (\text{Motore}, \text{Oggetto}) \}$  ;
  - $\text{ha-un} = \{ (\text{Automobile}, \text{Motore}) \}$  ;
  - $\text{ripara} = \{ (\text{Meccanico}, \text{Automobile}) \}$  ;
- $A = \{ \text{"}\forall a \in \text{Automobile } \exists m \in \text{Meccanico} : \text{ripara}(p, a)\text{"} \}$  ;

**Definizione** L’Ontology Mapping tra due ontologie consiste, in termini matematici, nell’applicarvi il processo di “*morfismo*”. Con *morfismo* si intende quel processo di “mapping” da una struttura matematica (come può essere, per l’appunto, un grafo) ad un’altra, preservandone l’integrità.

In parole povere consiste in una serie di funzioni che assegnano ai simboli usati in un vocabolario i simboli di un altro vocabolario. Date due ontologie,  $O_1 = (G_1, A_1)$  a  $O_2 = (G_2, A_2)$ , possiamo distinguere due tipologie di ontology mapping [5]:

- ontology mapping totale
- ontology mapping parziale

**Ontology mapping totale** L’ontology mapping totale da  $O_1$  a  $O_2$  è un morfismo  $f : G_1 \rightarrow G_2$  tale che  $A_2 \models f(A_1)$ , cioè tutte le interpretazioni che soddisfano gli assiomi di  $O_2$  soddisfano anche gli assiomi “tradotti” di  $O_1$ .

**Ontology Mapping parziale** Esiste un ontology mapping parziale da  $O_1$  a  $O_2$  se e solo se esiste una sub-ontologia  $O'_1 = (G'_1, A'_1)$  (con  $G'_1 \subseteq G_1$  e  $A'_1 \subseteq A_1$ ) tale che ci sia un mapping totale da  $O'_1$  a  $O_2$ .

## 2.6 Answer Set Programming (ASP)

**Definizione** L’Answer Set Programming (ASP), spesso utilizzata per la risoluzione di problemi computazionalmente difficili come la colorazione dei grafi, il cammino hamiltoniano o la soddisfacibilità booleana, è una forma di *programmazione logica* che segue il paradigma della *programmazione dichiarativa*, spostando il focus sulla dichiarazione di un problema piuttosto che sulla formulazione di algoritmi per risolverlo (compito svolto dal software di supporto), usando quindi un approccio antitetico rispetto a quello della *programmazione imperativa*.

Il fine ultimo è quello di trovare un “*modello stabile*” (*stable model*), o “*answer set*”, che funga da soluzione al problema dato.

La scrittura di un programma in ASP per la risoluzione di un problema genericamente consiste nello scrivere un insieme di *atomi*, che definiscono la

“base di conoscenza” del dominio (*knowledge base*), ed un insieme di *regole* e *constraint*, che descrivono gli assiomi, vincoli e condizioni del problema dato.

**Esempio** Senza scendere nel dettaglio sulla sintassi della programmazione ASP, di seguito è riportato un esempio di risoluzione in ASP per il problema della  $\beta$ -colorabilità.

La  $n$ -colorabilità di un grafo  $G = (V, E)$  è una funzione  $color : V \rightarrow 1..n$  tale che  $color(x) \neq color(y)$  per ogni coppia di vertici adiacenti  $(x, y) \in E$ . Nell'esempio userò “stato(X)” per indicare un nodo  $X$  e “confine(X,Y)” per indicare un arco che collega il nodo  $X$  con il nodo  $Y$ .

Il programma in ASP risultante è il seguente:

```
% nodi
stato(a). stato(b). stato(c). stato(d).

% archi
confine(a,b). confine(b,a). confine(a,c).
confine(c,a). confine(c,d). confine(d,c).

% regola per colorare il nodo “X” con uno dei colori
colora(X,”rosso”)|colora(X,”verde”)|colora(X,”blu”):- stato(X).

% vincolo che impedisce ad una coppia di vertici
% adiacenti di avere lo stesso colore
:- confine(X,Y), colora(X,C), colora(Y,C).
```

Da questo esempio si evince la potenzialità dell'ASP in contesti semantici.

# Capitolo 3

## Ontology Mapping con ASP

### 3.1 Procedimento

Di seguito verrà mostrato un esempio di *ontology mapping parziale* (def. nel par. 2.5) dall'ontologia “DBpedia Italiana” a quella “CIDOC-CRM” relativo solo a concetti e relazioni riguardanti le opere d'arte . Lo scopo è quello di creare un file contenente le risorse di DBpedia Italiana rappresentate da concetti e relazioni di CIDOC-CRM. Il procedimento si può riassumere in 5 fasi:

1. reperimento file contenenti le risorse DBpedia Italiana
2. estrazione statement relativi alle opere d'arte
3. studio delle due ontologie
4. creazione delle regole di mapping
5. creazione file con tutte le risorse mappate

**Fase 1** Riguardo questa fase non c'è molto da dire. I data set che contengono le informazioni di DBpedia Italiana sono stati scaricati dal sito “<http://it.dbpedia.org/dati/>” in formato Turtle (estensione “.ttl”).

```

# started 2015-01-29T09:18:58Z
itdb:resource/Gioconda dbo:imageSize "280"^^xsd:integer .
itdb:resource/Gioconda foaf:name "La Gioconda"@it .
itdb:resource/Gioconda dbo:author itdb:resource/Leonardo_da_Vinci .
itdb:resource/Gioconda dbo:technique "olio su tavola"^^xsd:string .
itdb:resource/Gioconda dbo:height "0.77"^^xsd:double .
itdb:resource/Gioconda dbo:width "0.53"^^xsd:double .
itdb:resource/Gioconda dbo:location itdb:resource/Parigi .
itdb:resource/Gioconda dbo:location itdb:resource/Musée_du_Louvre .
itdb:resource/Statua_di_Zeus_a_Olimpia foaf:name "Statua di Zeus a Olimpia"@it .
itdb:resource/Statua_di_Zeus_a_Olimpia dbo:author itdb:resource/Fidia .
itdb:resource/Statua_di_Zeus_a_Olimpia dbo:material "oro e avorio"^^xsd:string .
itdb:resource/Statua_di_Zeus_a_Olimpia dbo:height "12.0"^^xsd:double .
itdb:resource/Statua_di_Zeus_a_Olimpia dbo:location itdb:resource/Olimpia .
itdb:resource/Nascita_di_Venere dbo:imageSize "350"^^xsd:integer .
itdb:resource/Nascita_di_Venere foaf:name "Nascita di Venere"@it .
itdb:resource/Nascita_di_Venere dbo:author itdb:resource/Sandro_Botticelli .
itdb:resource/Nascita_di_Venere dbo:technique "tempera su tela"^^xsd:string .
itdb:resource/Nascita_di_Venere dbo:height "1.72"^^xsd:double .
itdb:resource/Nascita_di_Venere dbo:width "2.78"^^xsd:double .
itdb:resource/Nascita_di_Venere dbo:location itdb:resource/Firenze .
itdb:resource/Nascita_di_Venere dbo:location itdb:resource/Galleria_degli_Uffizi .
itdb:resource/Primavera_(Botticelli) dbo:imageSize "400"^^xsd:integer .
itdb:resource/Primavera_(Botticelli) foaf:name "Primavera"@it .
itdb:resource/Primavera_(Botticelli) dbo:author itdb:resource/Sandro_Botticelli .
itdb:resource/Primavera_(Botticelli) dbo:technique "Tempera su tavola"^^xsd:string .

```

Figura 3.1: Esempio di alcuni statement contenuti nei data set di DBPedia Italiana

```

2 | itdb:resource/Gioconda rdf:type dbo:Artwork .

```

Figura 3.2: Statement RDF che categorizza la risorsa *Gioconda* come opera d’arte.

**Fase 2** Vista l’intenzione di fare un ontology mapping parziale, limitato al dominio delle opere d’arte, sono stati estratti dai file solo gli statement d’interesse. Per fare ciò si sono anzitutto identificate le risorse di interesse (sculture, quadri, ecc...). Queste, in DBPedia, sono identificabili tramite statement che hanno come proprietà “<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>” e come oggetto “<http://dbpedia.org/ontology/Artwork>” [Fig.3.2]. Una volta identificate le opere d’arte, applicando un algoritmo di chiusura transitiva, sono state estratte anche tutte le risorse ad esse collegate.

**Fase 3** Questa fase è dedicata allo studio approfondito delle due ontologie, analizzando in particolar modo le differenze nella rappresentazione di concetti e relazioni simili. Per fare ciò, si è andati alla ricerca di statement

semanticamente equivalenti da poter confrontare, rappresentati sia dal modello DBpedia Italiana che da quello CIDOC-CRM. Questa fase è cruciale per permettere la creazione di regole di mapping che mantengano inalterata la semantica dei dati.

**Fase 4** Ultimate le precedenti fasi, sono state acquisite abbastanza informazioni per poter iniziare la fase di ontology mapping. Come detto in precedenza, per mappare le due ontologie è stato fatto uso della programmazione ASP. Anzitutto è stato creato un file di testo chiamato “*facts\_file.txt*”. Poi è stato definito il dominio, utilizzando uno script [Alg. 1] per la creazione dei fatti ASP rappresentanti tutte le triple RDF contenute nei file DBpedia precedentemente scaricati e puliti. Ogni tripla RDF è stata rappresentata da un fatto ASP del tipo [3]:

*tripla(“Soggetto”, “Predicato”, “Oggetto”).*

ed è stata poi inserita nel file “*facts\_file.txt*”. [Fig.3.3]

```
Data: File facts_file, List<File>filesDBPedia;
foreach File f ∈ filesDBPedia do
    foreach Statement s ∈ f do
        write tripla(“+s.Soggetto+”, “+s.Predicato+”, “s.Oggetto”).
        in facts_file;
    end
end
```

**Algorithm 1:** scrittura fatti ASP rappresentanti le triple RDF di DBpedia nel file “*facts\_file.txt*”

Definito il dominio, è stato creato il file “*mapping\_rules.txt*” dove sono state scritte le regole necessarie al mapping dei concetti e delle relazioni [Fig. 3.4]. Per mappare alcune informazioni, c’è stata la necessità di creare nuovi URI. La radice usata per la creazione dei nuovi URI è:

“<http://mat.unical.it/SemanticWebProject/>”.

```

1 tripla("itdb:resource/Guernica_(Picasso)","foaf:name","Guernica@it").
2 tripla("itdb:resource/Guernica_(Picasso)","dbo:author","itdb:resource/Pablo_Picasso").
3 tripla("itdb:resource/Guernica_(Picasso)","dbo:technique","olio su tela^^xsd:string").
4 tripla("itdb:resource/Guernica_(Picasso)","dbo:height","3.49^^xsd:double").
5 tripla("itdb:resource/Guernica_(Picasso)","dbo:width","7.76^^xsd:double").
6 tripla("itdb:resource/Guernica_(Picasso)","dbo:location","itdb:resource/Madrid").
7 tripla("itdb:resource/Guernica_(Picasso)","dbo:location","itdb:resource/Museo_Nacional_Centro_de_Arte_Reina_Sofia").
8 tripla("itdb:resource/Modena","foaf:name","Modena @it").
9 tripla("itdb:resource/Modena","dbo:administrativeDistrict","itdb:resource/Emilia-Romagna").
10 tripla("itdb:resource/Modena","dbo:administrativeDistrict","itdb:resource/Stazione_di_Modena").
11 tripla("itdb:resource/Modena","dbo:leaderParty","itdb:resource/Partito_Democratico").
12 tripla("itdb:resource/Modena","dbo:electionDate","2014-06-09^^xsd:date").
13 tripla("itdb:resource/Modena","rdf:type","http://www.opengis.net/gml/_Feature").
14 tripla("itdb:resource/Modena","geo:lat","44.64472222222222^^xsd:float").
15 tripla("itdb:resource/Modena","geo:long","10.925555555555556^^xsd:string").
16 tripla("itdb:resource/Modena","grs:point","44.64472222222222 10.925555555555556^^xsd:string").
17 tripla("itdb:resource/Modena","dbo:elevation","34.0^^xsd:double").
18 tripla("itdb:resource/Modena","dbo:area","1.83E8^^xsd:double").
19 tripla("itdb:resource/Modena","dbo:populationTotal","184915^^xsd:nonNegativeInteger").
20 tripla("itdb:resource/Modena","dbo:populationAsOf","2014-04-30^^xsd:date").
21 tripla("itdb:resource/Modena","dbo:postalCode","41121-41126^^xsd:string").
22 tripla("itdb:resource/Modena","dbo:areaCode","059^^xsd:string").
23 tripla("itdb:resource/Modena","dbo:vehicleCode","MO^^xsd:string").
24 tripla("itdb:resource/Modena","dbo:demonym","Modenesi o Geminiani @it").
25 tripla("itdb:resource/Modena","dbo:saint","itdb:resource/San_Geminiano").
26 tripla("itdb:resource/Modena","foaf:homepage","http://www.comune.modena.it").
27 tripla("itdb:resource/Idolino_di_Pesaro","foaf:name","Idolino di Pesaro @it").
28 tripla("itdb:resource/Idolino_di_Pesaro","dbo:technique","bronzo^^xsd:string").
29 tripla("itdb:resource/Idolino_di_Pesaro","dbo:height","1.46^^xsd:double").
30 tripla("itdb:resource/Idolino_di_Pesaro","dbo:location","itdb:resource/Firenze").
31 tripla("itdb:resource/Idolino_di_Pesaro","dbo:location","itdb:resource/Museo_archeologico_nazionale_di_Firenze").
32 tripla("itdb:resource/Stele_di_Avile_Tite","foaf:depiction","itdb:resource/Museo_Guarnacci").
33 tripla("itdb:resource/Stele_di_Avile_Tite","dbo:imageSize","200^^xsd:integer").
34 tripla("itdb:resource/Stele_di_Avile_Tite","foaf:name","Stele di Avile Tite @it").
35 tripla("itdb:resource/Stele_di_Avile_Tite","dbo:height","1.7^^xsd:double").
36 tripla("itdb:resource/Stele_di_Avile_Tite","dbo:location","itdb:resource/Volterra").
37 tripla("itdb:resource/Stele_di_Avile_Tite","dbo:location","itdb:resource/Museo_Guarnacci").
38 tripla("itdb:resource/Urneta_con_scena_di_banchetto","foaf:name","Urneta con scena di banchetto @it").
39 tripla("itdb:resource/Urneta_con_scena_di_banchetto","dbo:material","pietra fetida chiusina^^xsd:string").
40 tripla("itdb:resource/Urneta con scena di banchetto","dbo:height","0.3^^xsd:double").

```

Figura 3.3: Porzione del file “facts\_file.txt”.

```

1%=====
2%===== <REGOLE PER IL MAPPING>
3%=====
4%===== <CATEGORIZZAZIONE DELLE RISORSE>
5 autore(A) :- tripla(O,"dbo:author",A).
6 opera(X) :- tripla(X,"rdf:type","dbo:Artwork").
7 istituzione(X) :- tripla(X,"rdf:type","dbo:Organisation") .
8 istituzione(X) :- tripla(X,"rdf:type","dbo:Museum") .
9 persona(X) :- tripla(X,"rdf:type","dbo:Person") .
10 posto(X) :- tripla(X,"rdf:type","dbo:Place"), not istituzione(X).
11%===== <REGOLE PER MAPPARE I TIPI>
12 triplaAS(X,"rdf:type","crm:E22_Man-Made_Object") :- opera(X) .
13 triplaAS(X,"rdf:type","crm:E40_Legal_Body") :- istituzione(X) .
14 triplaAS(X,"rdf:type","crm:E21_Person") :- persona(X) .
15 triplaAS(X,"rdf:type","crm:E53_Place") :- posto(X).
16%===== <REGOLE PER MAPPARE VARI CONCETTI>
17% tecnica utilizzata per la creazione di un'opera
18 triplaAS(0, "crm:P32_used_general_technique", T) :- tripla(0, "dbo:technique", T).
19
20% materiale di cui è composta un'opera
21 triplaAS(L,"crm:P45_consists_of",M):- tripla(L,"itdb:property/materiale", M).
22 triplaAS(M,"rdf:type","crm:E57_Materials"):- tripla(L,"itdb:property/materiale", M).
23
24% istituzione che custodisce attualmente l'opera
25 triplaAS(0,"crm:P50_has_current_keeper",L) :- tripla(0, "dbo:location", L), istituzione(L).
26
27% posto (inteso come luogo geografico) dove l'opera si trova attualmente
28 triplaAS(0,"crm:P55_has_current_location",L) :- tripla(0, "dbo:location", L), not istituzione(L).
29
30% immagini dell'opera
31 triplaAS(0,"crm:P138i_has_representation",I):- tripla(0,"foaf:depiction",I).
32 triplaAS(0,"crm:P138i_has_representation",I):- tripla(0,"dbo:thumbnail",I).
33%===== <REGOLE PER MAPPARE IL CONCETTO DI "TITOLO">
34 #const urititoli = "[URI-TITLE]" .
35
36 triplaNewURI(0, "crm:P102_has_title", urititoli) :- tripla(0,"foaf:name", T), opera(O).
37 quadruplaOUUU(0, urititoli, "rdf:type", "crm:E35_Title") :- triplaNewURI(0, "crm:P102_has_title", urititoli).
38 quadruplaOUUS(0, urititoli, "rdfs:label", T) :- triplaNewURI(0, "crm:P102_has_title", urititoli), tripla(0, "foaf:name", T), opera(O).
39%===== <REGOLE PER MAPPARE IL CONCETTO DI "AUTORE">
40% costante utile in fase di post-processing per identificare triple di "produzione" (intesa come concetto CIDOC-CRM)
41 #const uriEventoProduzione = "[URI-PRODUZIONE]" .

```

Figura 3.4: Alcune regole di mapping contenute nel file “mapping\_rules.txt”.

```

1 @prefix uswp: <http://mat.unical.it/SemanticWebProject/>
2 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 @prefix crm: <http://erlangen-crm.org/current/>
4 @prefix itdb: <http://it.dbpedia.org/>
5
6 uswp:Gioconda/production rdf:type crm:E12_Production .
7 uswp:Gioconda/production crm:P14_carried_out_by itdb:resource/Leonardo_da_Vinci .
8 uswp:Gioconda/production crm:P4_has_time-span uswp:Gioconda/production/date .
9 uswp:Gioconda/production/date rdf:type crm:E52_Time-Span .
10 uswp:Gioconda/dimension/height rdf:type crm:E54_Dimension .
11 uswp:Gioconda/dimension/width rdf:type crm:E54_Dimension .
12 uswp:Gioconda/title rdf:type crm:E35_Title .
13 uswp:Statua_di_Zeus_a_Olimpia/production rdf:type crm:E12_Production .
14 uswp:Statua_di_Zeus_a_Olimpia/production crm:P14_carried_out_by itdb:resource/Fidia .
15 uswp:Statua_di_Zeus_a_Olimpia/production crm:P4_has_time-span uswp:Statua_di_Zeus_a_Olimpia/production/date .
16 uswp:Statua_di_Zeus_a_Olimpia/production/date rdf:type crm:E52_Time-Span .
17 uswp:Statua_di_Zeus_a_Olimpia/dimension/height rdf:type crm:E54_Dimension .
18 uswp:Statua_di_Zeus_a_Olimpia/title rdf:type crm:E35_Title .
19 uswp:Nascita_di_Venere/production rdf:type crm:E12_Production .
20 uswp:Nascita_di_Venere/production crm:P14_carried_out_by itdb:resource/Sandro_Botticelli .
21 uswp:Nascita_di_Venere/production crm:P4_has_time-span uswp:Nascita_di_Venere/production/date .
22 uswp:Nascita_di_Venere/production/date rdf:type crm:E52_Time-Span .
23 uswp:Nascita_di_Venere/dimension/height rdf:type crm:E54_Dimension .
24 uswp:Nascita_di_Venere/dimension/width rdf:type crm:E54_Dimension .
25 uswp:Nascita_di_Venere/title rdf:type crm:E35_Title .
26 uswp:Primavera_(Botticelli)/production rdf:type crm:E12_Production .
27 uswp:Primavera_(Botticelli)/production crm:P14_carried_out_by itdb:resource/Sandro_Botticelli .
28 uswp:Primavera_(Botticelli)/production crm:P4_has_time-span uswp:Primavera_(Botticelli)/production/date .
29 uswp:Primavera_(Botticelli)/production/date rdf:type crm:E52_Time-Span .
30 uswp:Primavera_(Botticelli)/dimension/height rdf:type crm:E54_Dimension .
31 uswp:Primavera_(Botticelli)/dimension/width rdf:type crm:E54_Dimension .
32 uswp:Primavera_(Botticelli)/title rdf:type crm:E35_Title .
33 uswp:David_(Michelangelo)/production rdf:type crm:E12_Production .
34 uswp:David_(Michelangelo)/production crm:P14_carried_out_by itdb:resource/Michelangelo_Buonarroti .
35 uswp:David_(Michelangelo)/production crm:P4_has_time-span uswp:David_(Michelangelo)/production/date .
36 uswp:David_(Michelangelo)/production/date rdf:type crm:E52_Time-Span .
37 uswp:David_(Michelangelo)/dimension/height rdf:type crm:E54_Dimension .
38 uswp:David_(Michelangelo)/dimension/width rdf:type crm:E54_Dimension .
39 uswp:David_(Michelangelo)/title rdf:type crm:E35_Title .

```

Figura 3.5: Alcuni degli statement ottenuti dal mapping

**Fase 5** A questo punto è stato utilizzato il file ASP precedentemente creato (fase 4), come input di un answer set solver (nel nostro caso DLV). Ricevuto in output il modello stabile è stato effettuato un parsing, estraendo da ogni fatto ASP “Soggetto”, “Predicato” e “Oggetto”, ottenendo così gli statement utilizzati per la creazione del file contenente il risultato finale della mappatura [Fig. 3.5]. Per alcune triple, però, non è stato possibile utilizzare esclusivamente regole ASP e fare un semplice parsing, ma è stato necessario ricorrere ad attività di post-processing per creare gli statement (come nel caso delle triple che richiedevano la creazione di nuovi URI).

# Capitolo 4

## Validazione

Per quanto concerne la validazione del lavoro svolto, essa è avvenuta tramite l'utilizzo dell'applicazione “*CHQA*” (Cultural Heritage Question Answering).

### 4.1 Applicazione CHQA

*CHQA* è un'applicazione che fornisce un sistema di *question answering* relativo a domande inerenti l'ambito dei beni culturali.

L'applicazione richiede, come prima cosa, l'inserimento di una domanda posta in linguaggio naturale.

Una volta posta, la domanda viene elaborata dal parser (*Stanford NLP*) che la suddivide in token ed esegue un'analisi grammaticale su di essi per identificare le “parti del discorso” (verbi, aggettivi, sostantivi, articoli, ecc...). A questo punto esso cerca di trovare un template che combaci con la domanda, tramite l'utilizzo dell'Answer Set Programming e del solver *DLV*.

Trovato il template, l'applicazione procede alla formulazione ed esecuzione di una query SPARQL sull'endPoint del British Museum (<https://collection.britishmuseum.org/>) per estrapolare le informazioni richieste.

Una volta che la query ha ottenuto una risposta, quest'ultima viene mostrata all'utente.

## 4.2 Validazione tramite CHQA

Per validare il lavoro è stata utilizzata CHQA supportata da un database, con supporto RDF, gestito dal tool “*OpenLink Virtuoso*”. Per validare il lavoro prima di tutto è stato importato nel database il file di output con estensione .ttl ottenuto dalla *Fase 5* (par. 3.1). Successivamente sono state poste una serie di domande all’applicazione per verificare la correttezza dell’output.

Esempio:

- **Domanda:** “*Chi dipinse Guernica?*”
- **Output desiderato:** “*Pablo Diego Josè Francisco de Paula Juan Nepomuceno María de los Remedios Cipriano de la Santísima Trinidad Ruiz y Picasso* ”
- **Principale query SPARQL eseguita da CHQA:**

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX crm: <http://erlangen-crm.org/current/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX itdb: <http://it.dbpedia.org/>
SELECT DISTINCT ?outputLabel ?output
WHERE {
    itdb:resource/Guernica_(Picasso) crm:P108i_was_produced_by
    ?var1 .
    ?var1 crm:P14_carried_out_by ?output .
    ?output a crm:E21_Person .
    ?output rdfs:label ?outputLabel .
}
```

- **Output ottenuto:**

Autore: Pablo Diego Josè Francisco de Paula  
Juan Nepomuceno Marìa de los Remedios  
Cipriano de la Santìsima Trinidad Ruiz y  
Picasso

- Altre informazioni disponibili su Pablo Diego Josè Francisco  
de Paula Juan Nepomuceno Marìa de los Remedios Cipriano  
de la Santìsima Trinidad Ruiz y Picasso –

Data di nascita: 1881

Data di morte: 1973

Opera realizzata: Acrobata e giovane equilibrista

Opera realizzata: Arlecchino

...

- Altre informazioni disponibili su Guernica –

Data di creazione: 1937

Luogo in cui è conservata l'opera: Madrid .

# **Capitolo 5**

## **Conclusioni**

In questo lavoro di tesi abbiamo affrontato il problema di Ontology Mapping e adottato una soluzione basata su Answer Set Programming che permettesse di effettuare il mapping in modo semplice e intuitivo tramite regole logiche. Nello specifico abbiamo verificato sperimentalmente la possibilità di applicare ASP su un esempio di Ontology Mapping dall'ontologia DBpedia Italiana (estensione dell'ontologia DBpedia) all'ontologia CIDOC-CRM.

L'approccio al problema tramite l'utilizzo dell'Answer Set Programming si è rilevato estremamente efficace. Le regole di mapping possono essere facilmente modificate ed adattate per fare il mapping tra due ontologie diverse da quelle utilizzate in questo lavoro. Infine abbiamo validato sperimentalmente il lavoro svolto servendoci di un sistema di Question Answering basato su CIDOC-CRM e abbiamo dunque verificato la possibilità di integrare i dati provenienti da dbpedia dentro il sistema, ampliandone di fatto la base di conoscenza.

# Bibliografia

- [1] Intelligent web, ontology and beyond. [http://www.ece.ualberta.ca/~reform/ece627w2013/lec11\\_mediation.ppt](http://www.ece.ualberta.ca/~reform/ece627w2013/lec11_mediation.ppt), 2013. [Online; accessed December-2017].
- [2] Tim Berners-Lee. *L'architettura del nuovo Web*. Feltrinelli, marzo 2001.
- [3] Thomas Eiter, Roman Schindlauer, Giovambattista Ianni, and Axel Polleres. Answer set programming for the semantic web: Tutorial. <https://www.mat.unical.it/asptut/slides/unit4.pdf>, 2006. [Online; accessed December-2017].
- [4] Alessandro Longheu. Dati semistrutturati: Ontologie. [http://www.ece.ualberta.ca/~reform/ece627w2013/lec11\\_mediation.ppt](http://www.ece.ualberta.ca/~reform/ece627w2013/lec11_mediation.ppt), 2009-2010. [Online; accessed December-2017].
- [5] Marco Schorlemmer Yannis Kalfoglou. Ontology mapping: The state of the art. *The Knowledge Engineering Review*, 18:1–31, January 2003.